PENDING CLAIMS AS AMENDED

Please amend the claims as follows:

1. (Currently Amended) In a communication system, a method for decoding a sequence of turbo encoded data symbols transmitted over a channel comprising:

updating channel nodes R_x, R_y and R_z based on a received channel output;

initializing outgoing messages from symbol nodes X_i, Y_i and Z_k, wherein said symbol nodes X_i, Y_i and Z_k are in communication with said channel nodes R_x, R_y and R_z; and

triggering updates of computational nodes C and D, associated with different instances of time, in accordance with a triggering schedule, wherein a computational node C_i is in communication with said symbol nodes X_i and Y_i and a computational node D_k is in communication with said symbol nodes X_i and [[Z_k.]] Z_k;

wherein said triggering schedule includes triggering all said computational nodes C and D at different instances of time essentially concurrently for each decoding iteration.

2. (Original) The method as recited in claim 1 wherein said computational node C_i is in communication with state nodes S_i and S_{i-1} associated with a first constituent code, and said computational node D_k is in communication with state nodes σ_k and σ_{k-1} associated with a second constituent code, wherein said first and second constituent codes are associated with a turbo code in said communication system used for encoding said sequence of encoded data symbols.

3. (Original) The method as recited in claim 1 further comprising:

accepting a value of symbol X_i at said symbol node X_i as a decoded value of symbol X_i after at least one iteration of said triggering updates of said computational nodes C and D.

4. (Canceled)

5. (Canceled)

6. (Currently Amended) In a communication system, a method for decoding a sequence of turbo encoded data symbols transmitted over a channel comprising:

updating channel nodes R_x, R_y and R_z based on a received channel output;

initializing outgoing messages from symbol nodes X_i, Y_i and Z_k, wherein said symbol nodes X_i, Y_i and Z_k are in communication with said channel nodes R_x, R_y and R_z;

triggering updates of computational nodes C and D, associated with different instances of time, in accordance with a triggering schedule, wherein a computational node C_i is in communication with said symbol nodes X_i and Y_i and a computational node D_k is in communication with said symbol nodes X_i and Z_k; and

~~The method as recited in claim 1 further comprising:~~

partitioning said computational node C at time instances C_0, C_1, C_2, ..., C_N into at least two subsets, wherein said triggering schedule includes triggering updates of computational nodes C in a sequence at different time instances in each subset, and wherein said triggering of computational node C at different time instances in said least two subsets occurs concurrently.

7. (Original) The method as recited in claim 6 further comprising:

determining said sequence at different time instances in each subset for said triggering updates.

8. (Canceled)

9. (Original) The method as recited in claim 6 wherein said least two subsets of computational node C at different time instances C_0, C_1, C_2, ..., C_N have at least one common computational node time instance.

10. (Currently Amended) The method as recited in claim ~~1~~ 6 further comprising:

partitioning computational node D at different time instances D_0, D_1, D_2, ..., D_N into at least two subsets, wherein said triggering schedule includes triggering computational nodes D at different time instances in a sequence in each subset.

11. (Original) The method as recited in claim 10 further comprising:

determining said sequence at different time instances in each subset for said triggering updates.

12. (Original) The method as recited in claim 10 wherein said triggering of computational node D at different time instance in said least two subsets occurs concurrently.

13. (Original) The method as recited in claim 10 wherein said subsets of computational node D at time instances D_0, D_1, D_2, ..., D_N have at least one common computational node time instance.

14. (Original) The method as recited in claim 1 wherein said updating includes summing incoming messages to produce an output message, and outputting said output message for updating.

15. (Previously Presented) The method as recited in claim 1 wherein said updating said channel nodes R_x, R_y and R_z based on said received channel output includes:

receiving at said channel node R_x said channel output associated with a symbol X_i;

receiving at said channel node R_y said channel output associated with a symbol Y_i;

receiving at said channel node R_z said channel output associated with a symbol Z_k;

passing from said channel node R_x a likelihood of said symbol X_i, based on said received channel output, to said symbol node X_i;

passing from said channel node R_y a likelihood of said symbol Y_i, based on said received channel output, to said symbol node Y_i; and

passing from said channel node R_z a likelihood of said symbol Z_k, based on said received channel output, to said symbol node Z_k.

16. (Original) The method as recited in claim 1 wherein said initializing outgoing messages from symbol nodes X_i, Y_i and Z_k includes:

passing a message from said symbol node X_i to said computational node C_i of said computational node C, wherein said message is based on a summation of incoming messages at said symbol node X_i;

passing a message from said symbol node X_i to said computational node D_k of said computational node D, wherein said message is based on a summation of incoming messages at said symbol node X_i;

passing a message from said symbol node Y_i to said computational node C_i, wherein said message is based on said likelihood of data symbol Y_i; and

passing a message from said symbol node Z_k to said computational node D_k, wherein said message is based on said likelihood of data symbol Z_k.

17. (Original) The method as recited in claim 1 wherein said sequence of data includes "N" number of symbols, wherein each symbol in said sequence is identified by either a subscript "i" or "k," and wherein said subscript "i" and "k" are references to time instances in the decoding process.

18. (Currently Amended) An apparatus for decoding a sequence of turbo encoded data symbols communicated over a channel comprising:

channel nodes R_x, R_y and R_z for receiving channel output;

symbol nodes X_i, Y_i and Z_k in communication with said channel nodes R_x, R_y and R_z;

state nodes S_i and S_{i-1} associated with a first constituent code in a turbo code;

state nodes [[σk]] $\underline{\sigma_k}$ and σ_{k-1} associated with a second constituent code in said turbo code;

a computational node C_i in communication with said symbol nodes X_i and Y_i;

a computational node D_k in communication with said symbol nodes X_i and Z_k, wherein said computational node C_i is in communication with said state nodes S_i and S_{i-1} and said computational node D_k is in communication with said state nodes σ_k and σ_{k-1};

a computational node C_{i+1} in communication with said state node S_i;

a computational node C_{i-1} in communication with said state node S_{i-1};

a computational node D_{K+1} in communication with said state node σ_k; and

a computational node D_{k-1} in communication with said state node [[σ_{K+1},]] σ_{k-1};

wherein computational nodes C and D at different time instances are configured for updates in accordance with [[a]] an update triggering schedule, said update triggering schedule including concurrent triggering of each node of a first plurality of said computational nodes C, and concurrent triggering of each node of a second plurality of computational nodes D.

19. (Canceled)

20. (Original) The apparatus as recited in claim 18, wherein said update triggering schedule includes triggering updates in a sequence in a partitioned computational nodes C_0, C_1, C_2, ..., C_N of at least two subsets and in a sequence in a partitioned computational nodes D_0, D_1, D_2, ..., D_N of at least two subsets.

21. (Previously Presented) The apparatus as recited in claim 18 wherein said sequence of turbo encoded data symbols includes "N" number of symbols, wherein each symbol in said sequence is identified by either a subscript "i" or "k" corresponding to the subscripts used for said state nodes and said computational nodes.

22. (Currently Amended) A processor configured for decoding a sequence of turbo encoded data symbols for communication over a channel comprising:

channel nodes R_x, R_y and R_z for receiving channel output;

symbol nodes X_i, Y_i and Z_k in communication with said channel nodes R_x, Ry and R_z;

state nodes S_i and S_{i-1} associated with a first constituent code in a turbo code;

state nodes σ_k and σ_{k-1} associated with a second constituent code in said turbo code;

a computational node C_i in communication with said symbol nodes X_i and Y_i;

a computational node D_k in communication with said symbol nodes X_i and Z_k, wherein said computational node C_i is in communication with said state nodes S_i and S_{i-1} and said computational node D_k is in communication with said state nodes σ_k and σ_{k-1};

a computational node C_{i+1} in communication with said state node S_i;

a computational node C_{i-1} in communication with said state node S_{i-1};

a computational node D_{K+1} in communication with said state node σ_k; and

a computational node D_{k-1} in communication with said state node [[σ_{K+1},]] σ_{k-1};

wherein computational nodes C and D at different time instances are configured for updates in accordance with [[a]] an update triggering schedule, said update triggering schedule including concurrent triggering of each node of a first plurality of said computational nodes C, and concurrent triggering of each node of a second plurality of computational nodes D.

23. (Original) The processor as recited in claim 22 wherein said update triggering schedule includes triggering updates of said computational nodes C and D in a sequence of C_0, C_1, C_2, …, C_N, C_{N-1}, C_{N-2}, C_{N-3}, … C_2, C_1, C_0, D_0, D_1, D_2, …, D_N, D_{N-1}, D_{N-2}, D_{N-3}, … D_2, D_1, D_0.

24. (Original) The processor as recited in claim 22 wherein said sequence of data includes "N" number of symbols, wherein each symbol in said sequence is identified by either a subscript "i" or "k" corresponding to the subscripts used for said state nodes and said computational nodes.

25. (Canceled)